International Journal of Research in Industrial Engineering

www.riejournal.com

Int. J. Res. Ind. Eng. Vol. 14, No. 1 (2025) 152–176.

Paper Type: Research Paper

Analyzing the Customer Purchase Data of an Online Shopping Store by Data Mining: A Real Case Study in Iran

Nima Moradi[1,*], Mosayeb Jalilian[2]

[1] Information and Systems Engineering, Concordia University, Montreal, QC, Canada; nima_moradi99@yahoo.com.

[2] Information Systems, Supply Chain Management & Decision Support Department, Neoma Business School, Rouen, France; mosayeb.jalilian@neoma-bs.fr.

Citation:

Received: ----- *Revised:* ----- *Accepted:*---	Moradi, N., & Jalilian, M. (2025). Analyzing the customer purchase data of an online shopping store by data mining: A real case study in Iran. *International journal of research in industrial engineering*, *14*(1), 152-176.

Abstract

Nowadays, online shopping plays a vital role in providing services and delivering goods to customers in the context of business intelligence and e-commerce. This research analyzes the customer purchase data of an Iranian online shopping company in Tehran. Among the available datasets provided by the company, 200 thousand records of one week of transactions have been selected for the present study. Several classification methods (i.e., Random Forest, gradient-boosted trees, K-Nearest Neighbor (KNN), Naïve Bayes, Kernel Naïve Bayes, and Neural Networks) and clustering approaches have been applied to discover the knowledge and patterns. The results show that before balancing the dataset, the KNN algorithm with K=5 is the best classification method among the existing methods. However, after balancing, gradient-boosted trees outperform the other classification methods. For clustering methods, the results show that the K-Means algorithm with K=3 is more efficient regarding the average within centroid distance for each cluster. Finally, concluding remarks and suggestions for future studies are stated.

Keywords: Online shopping, Data mining, Classification, Clustering.

1 | Introduction

Online shopping stores play an important role in providing services and delivering goods to customers quickly, rapidly, and effectively. Due to high demand fluctuation, recognizing potential customers and keeping current customers while dealing with the market's demands is challenging for every online shopping company. One way to deal with the challenges of online shopping is to study the market and analyze past data to develop

Corresponding Author: nima_moradi99@yahoo.com

https://doi.org/10.22105/riej.2024.468414.1458

models and recognize hidden patterns. Knowing the patterns and finding the knowledge behind a large dataset, one helpful approach is data mining tools such as classification and clustering [1].

Data mining tools benefit actual online stores by helping analyze customer purchase data and derive valuable insights. Classification allows stores to group customers based on predefined labels, such as high-value, frequent, or at-risk customers, enabling personalized marketing strategies and loyalty programs. Conversely, clustering groups customers based on their purchasing behaviors without predefined categories, revealing hidden patterns or trends. This helps identify customer segments, such as those with similar preferences or buying habits, and tailor product recommendations or promotions accordingly. Both techniques improve decision-making, boost sales, and enhance customer satisfaction by offering a more customized shopping experience.

Several data-mining approaches have been proposed in the literature to analyze customer purchase behavior. These works could be categorized from the methodologies perspective suggested for customer purchase data analysis in the literature: 1) clustering techniques, 2) classification methods, 3) association rule mining, 4) hybrid models, and 5) market basket analysis. The papers related to each category are examined in the following.

1.1|Customer Segmentation through Clustering Techniques

Several studies have employed clustering algorithms to segment customers based on purchasing behavior. For example, Chang et al. [2] proposed an anticipation model for potential customers' purchasing behavior. Their proposed model was based on the past purchasing behavior of loyal customers and the web server log files of loyal and potential customers by applying clustering and association rules analysis. Also, in their work, clustering analysis was used to gather critical characteristics of loyal customers' personal information. Then, association rules analysis extracted knowledge of loyal customers' purchasing behavior to detect potential customers' near-future interest in a star product. Similarly, Kim and Hong [3] separated online customers regarding purchasing and the customers' intention to purchase on the web. They combined the Self-Organized Map (SOM) clustering results and the K-Means algorithm into a single model.

In addition, Ponyiam and Arch-int [4] suggested an improved method for examining the purchasing behavior of specific customers to obtain detailed data on their buying trends. The process consisted of three steps, initially, products were grouped by customer type using K-Means clustering, and then appropriate clusters were chosen using the Elbow method. The result of this phase was identical purchased product items, each grouped separately; in the next step, purchasing patterns were studied using the Apriori algorithm. Next, they defined the threshold two values in ARA-1. The results of these phases were the purchasing behaviors of the particular group. The experts evaluated the accuracy of purchasing product patterns in the third stage. The investigation involved testing the proposed approach with purchase data from a retail outlet in Thailand, indicating that the method could analyze purchasing patterns accurately using dimensional data. ARA-2 had an accuracy of over 88%, higher than ARA-1 with 38% accuracy.

Moreover, Liao et al. [5] integrated online shopping and home delivery with association rules to find unknown bundling of fresh and non-fresh products in a hypermarket. Also, the customers were segmented into clusters by clustering analysis, and the catalog was designed according to each cluster's consumption preferences. Via such a model, it is expected to attract more customers, open broader markets, and earn higher profits for hypermarkets. Amine et al. [6] studied a case study of applying the Length, Recency, Frequency, and Monetary (LRFM) model and clustering techniques for Moroccan e-commerce websites. They developed effective marketing strategies and adopted the LRFM model using a two-stage clustering method. In the first stage of the clustering method, the self-organizing maps method is used to find the best number of clusters and the initial centroid. Next, in the second stage, the K-Means method divides the customers into nine clusters according to their L, R, F, and M values. These studies show that clustering helps discover hidden patterns in customer data.

1.2 | Classification Methods for Customer Behavior Prediction

Several classification techniques have been utilized to predict customer behavior and optimize marketing strategies. In several works of literature, customer behavior toward online shopping is studied, such as [7–9]. Chen et al. [10] applied data mining techniques in customer-centric business intelligence for an online retailer. According to the RFM model, the customers were segmented into groups using the K-Means clustering algorithm and decision tree induction. Also, in a recent work, Alghanam et al. [11] suggested a data mining approach to improve prediction accuracy and discover association rules for "frequent item sets." The K-Means clustering algorithm was utilized to decrease the dataset size and improve the runtime of the proposed model—the suggested model employed four distinct classifiers, including C4.5, J48, CS-MC4, and MLR. Also, the Apriori algorithm suggested items by analyzing past purchases. The model was tested on the Northwind trader's dataset and achieved an accuracy of 95.2% with 8 clusters.

Moreover, Moon et al. [12] studied online shopping to find out the behavior of online shopping and customer satisfaction. They believed that the quality of the product, the price of the product compared to the local market, the policy of return, and the timely delivery of the product are essential elements of online shopping. As the methodology, they applied Naïve Bayes, Apiorir, Decision Tree, and Random Forest classification algorithms for the analysis. These studies highlight that classification methods effectively categorize customers based on historical behaviors and forecast future purchases. However, they often need to account for the dynamic nature of online shopping environments, where customer preferences evolve rapidly.

Kazemi et al. [13] examined the practical application of data mining in identifying potential customers and their criteria in a competitive business setting. It also describes recognizing potential customers who can become actual customers. After examining the baskets of items that customers purchase, a pattern is revealed due to the identified parameters. Using the decision tree tool, they pinpointed the primary criteria and its various sub-criteria and assessed their level of importance. They recognized their significance in analyzing customers' shopping baskets and converting potential customers into actual ones. According to the suggested model, it was recommended that organizations boost their spending on customers with the potential to become loyal.

1.3 | Association Rule Mining for Purchasing Patterns

A significant body of literature uses association rule mining to discover purchasing patterns. Anbalagan et al. [14] proposed a method as an updating technique to extract the association rules for inter and intra-transactions. Also, a cluster analysis was used to verify that the associated objects fall in the nominal cluster. They claimed that the results could be used to develop a well-structured shop to help customers choose related products by predicting them and providing customer satisfaction. Suchacka and Chodak [15] evaluated the purchase probability, categories of viewed products, and session features by applying association rule mining for actual online bookstore data. Their findings show the differences in high purchase probability for the customer types. Khasanah [16] segmented customers by Agglomerative Hierarchical Clustering (AHC) and found customer buying patterns using association rule mining. Their result was found in a mobile shop in Sleman Yogyakarta. The results showed that the most significant customer segment of the shop was male university students who came on weekends. Also, based on the association rule mining, it was observed that tempered glass, smartphones, action cameras, waterproof monopods, and power banks have strong relationships.

Furthermore, Riaz et al. [17] and Liao et al. [5] found that association rules can detect patterns that help design marketing strategies and product bundling. Riaz et al. [17] used intelligent association rule mining, clustering, and concept hierarchy to extract interesting shopping patterns from data. They claimed their analysis could help the online retailer set precise and efficient marketing strategies. However, while association rules help detect static product relationships, they must often be more adaptable to rapidly changing customer preferences.

1.4|Hybrid Models for Enhanced Prediction Accuracy

Some studies have proposed combining clustering and classification for more accurate customer behavior prediction. For instance, Chen et al. [10] used the K-Means clustering algorithm alongside decision tree induction for customer segmentation based on the RFM model. Similarly, Alghanam et al. [11] applied a combination of the K-Means clustering algorithm and various classifiers, including C4.5, J48, and CS-MC4, to improve prediction accuracy and runtime performance, achieving 95.2% accuracy. These hybrid approaches demonstrate the advantages of integrating multiple algorithms to capture diverse aspects of customer behavior. However, these models often focus on enhancing accuracy only after considering computational efficiency in large-scale, real-world datasets.

1.5|Market Basket and Transactional Data Analysis

Analyzing transactional data to uncover purchasing trends is another popular approach. Hidayat et al. [18] analyzed the market basket to know customers' attitudes by analyzing data from sales transactions. Their result was from a data analysis test using monthly sales transaction data of cosmetics in the Breiliant store during November 2018 with 34 data sales transactions. The results showed that a combination of products with strong support and confidence were Original Liquid Bleaching Seeds, Harva Peeling Gel, and Castor oil. Similarly, Samboteng et al. [19] utilized the market-based analysis method to examine every piece of data and establish patterns for each. A market-based analysis approach used an association rule with an a priori algorithm. This algorithm generated sales transactions with robust links between items in the transaction. It serves as a sales recommendation to assist users in receiving suggestions when they view the purchased itemset details. The experiments in the research showed that higher values of "minimum support" and "minimum confidence" lead to quicker recommendation generation with fewer suggestions. These studies show the utility of market basket analysis in understanding customer preferences. However, the use of these methods in real-time online settings, especially for rapidly evolving e-commerce platforms, still needs to be explored.

Moreover, data mining can also be used to find valuable markets for an online Customer Relationship Management (CRM) marketing strategy [20]. The coffee shop industry in Taiwan was studied in a real case study by Chiang [21]. Their approach used a fuzzy clustering algorithm and an apriori algorithm to analyze customers to find more marketing and purchasing knowledge of online CRM systems. Their research also found three challenging markets: one fuzzy market, two association rules, and two fuzzy association rules. Also, Riaz et al. [17] applied intelligent rule mining and clustering techniques to uncover shopping patterns, which could guide precise CRM marketing strategies.

The literature review shows that more studies and analyses are still needed to extract the knowledge and patterns from online shopping stores due to everyday changes in customer behavior towards online shopping experiences. The main contribution of the present work is to study and analyze actual customer purchase data for an Iranian online shopping company via various classification and clustering methods. The applied methods aim at finding the patterns and discovering the hidden knowledge behind this data.

The paper is organized as follows. Section 2 describes the collected data in more detail. Section 3 explains and implements the proposed methodology based on actual data. Section 4 shows the results and patterns obtained from the dataset. Finally, Section 5 presents the concluding remarks and suggestions for future studies.

2|Data Collection and Description

The Iranian e-commerce company whose data are used in this research is in Tehran (its name and title are not disclosed due to confidential considerations). This company has 30 million monthly visitors and receives 17.2 million visits daily. Alexa also ranks it as Iran's third most visited website (after Google and Varzesh3). The company was reportedly valued at $500 million in late 2015. According to statistics published until early 2021, more than 4 million different goods have been offered for sale by nearly 160,000 sellers. It can process

600,000 goods per day. Also, it has an open data program to provide a suitable platform for the research and development of data-driven sciences. In the open innovation program, the actual information of the transactions of more than 2 million customers and one hundred thousand goods are sampled. After many checks and hours of verification, cleaning by the technology and product team, and ensuring non-disclosure of confidential customer information, these data are available to the public. This data is a good infrastructure for conducting scientific research or developing data-driven products and technologies such as artificial intelligence, deep learning, machine learning, social and cognitive sciences, and other scientific fields. The available datasets are: 1) product history: these data include one hundred thousand samples of products and their sellers' prices, 2) user comments: this data contains one hundred thousand samples of user comments that include several comments for a product, 3) customer purchase history: this data includes 100,000 customer purchases that, like other digital data, are anonymous to protect customer privacy. This data has time and location, 4) quality product reviews: this data contains the history of more than one hundred thousand products, and 5) product list: this data includes one hundred thousand samples of products and their classification.

2.1 | Data Visualization

Data visualization, which could be interpreted as representing information and data graphically, uses visual elements like maps, charts, and graphs to provide an easy way of understanding patterns, trends, and outliers. This section uses visual elements to understand better what is happening among the data. Dataset 3, customer purchase history, is chosen since the methodologies are implemented over this dataset. This set consists of 200,000 rows and different columns. *Fig. 1* shows the distribution of various instances of prices. Most purchases include products with a low price, say less than 50 million Rials. A few records of products that cost more than 150 million Rials could be considered outliers if the price is regarded as an attribute.

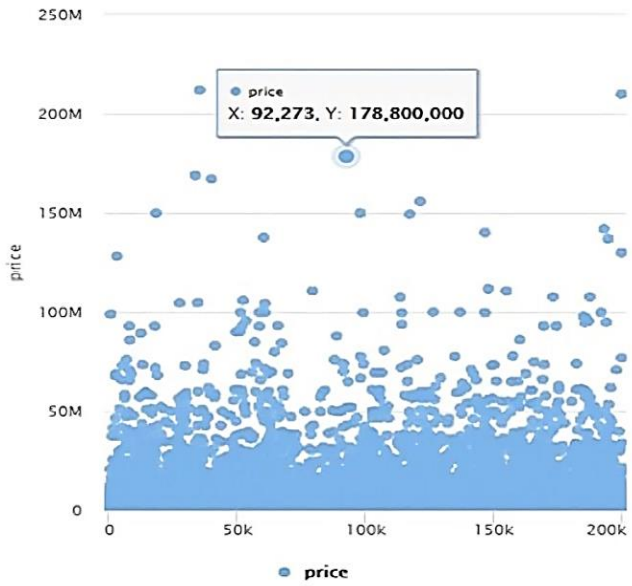

Fig. 1. Different values for price in all instances (transactions) in dataset 3.

Moreover, *Fig. 2* shows the purchase of products with different prices. The only thing that might be interesting is that high-quality products are purchased primarily in the afternoon. Also, *Fig. 3* indicates that the quantity of almost all the purchased products prices by the customers is less than 5. It is seen that just for a few products with the lowest prices, the purchased quantity is more than 25. Therefore, the more expensive a product is, the less demand for it. In *Fig. 4*, most of the products purchased by the customers have a price of less than 50 million Rials. In addition, *Fig. 5* represents the number of daily and time transactions. Our dataset is for a short period, which is one week, and for having a precise analysis of the times of the purchase and their correlations with the number of transactions, we should have the data for a more extended period, like a month or more.

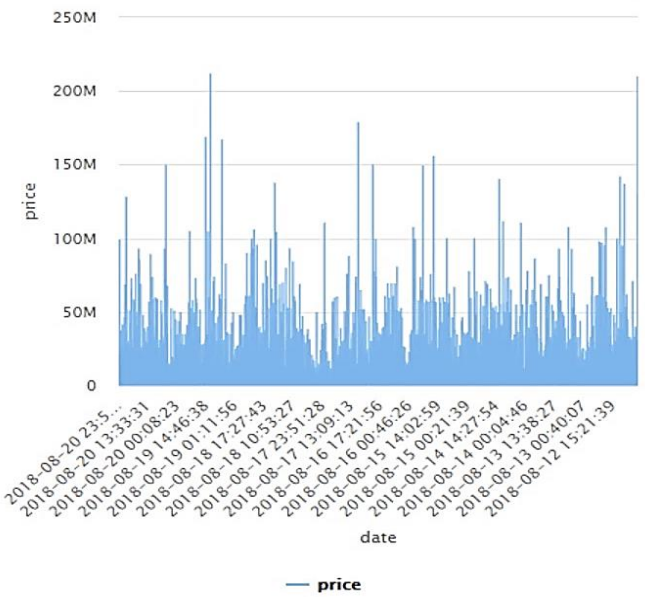

Fig. 2. The prices of each transaction at each day and time in dataset 3.

One valuable visualization is *Fig. 6*, showing the number of transactions that different users have performed. We can use this information to further work on customer segmentation or customer loyalty analysis. This work finds the frequency of different purchased products from *Fig. 7*. The frequency of just one product has reached 2000 times, and a few products have a frequency of more than 1000. Most of the products have a frequency of less than 300.

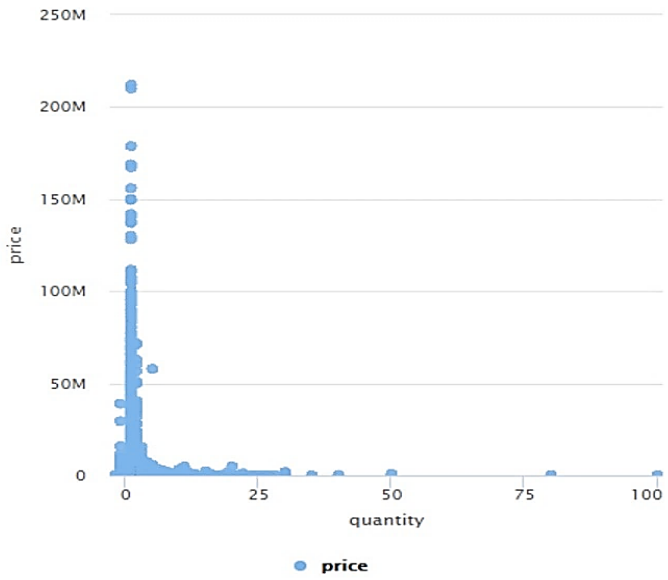

Fig. 3. Prices vs. quantities in data set 3.

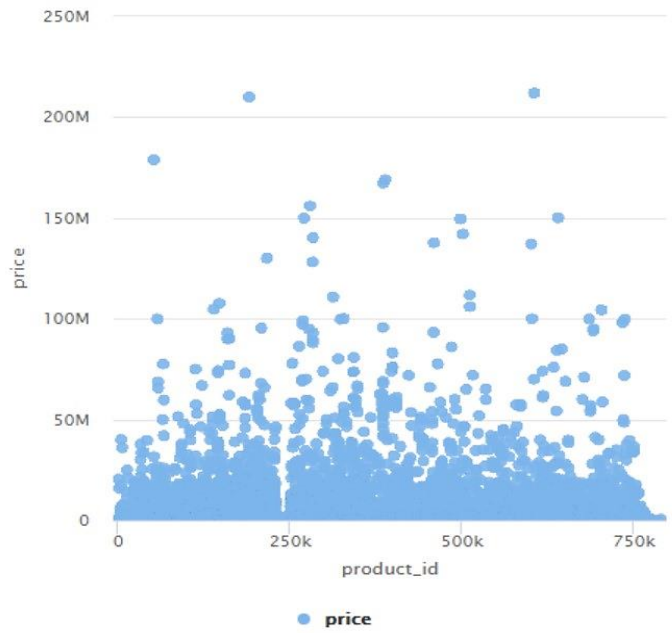

Fig. 4. Different values for price for each product in data set 3.

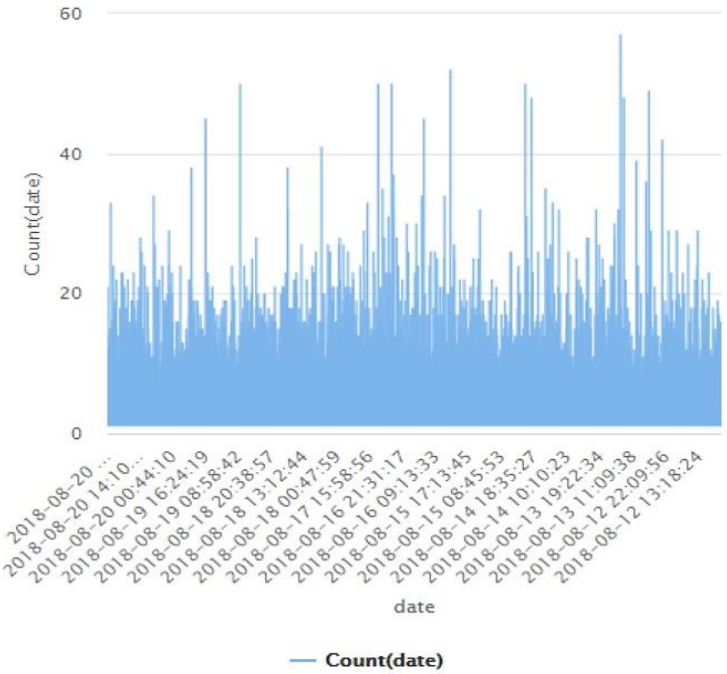

Fig. 5. Frequency of transactions at each day and time in data set 3.

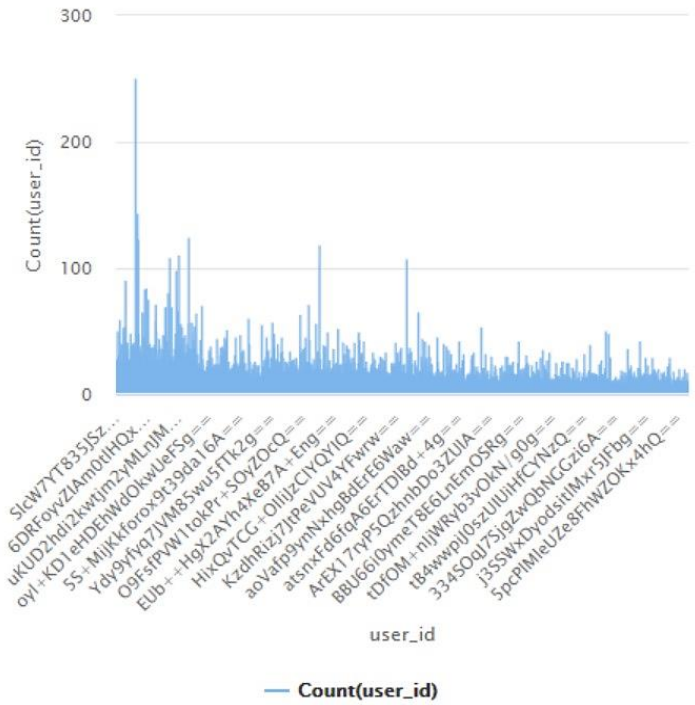

Fig. 6. Frequency of transactions made by each user (customer) in the data set.

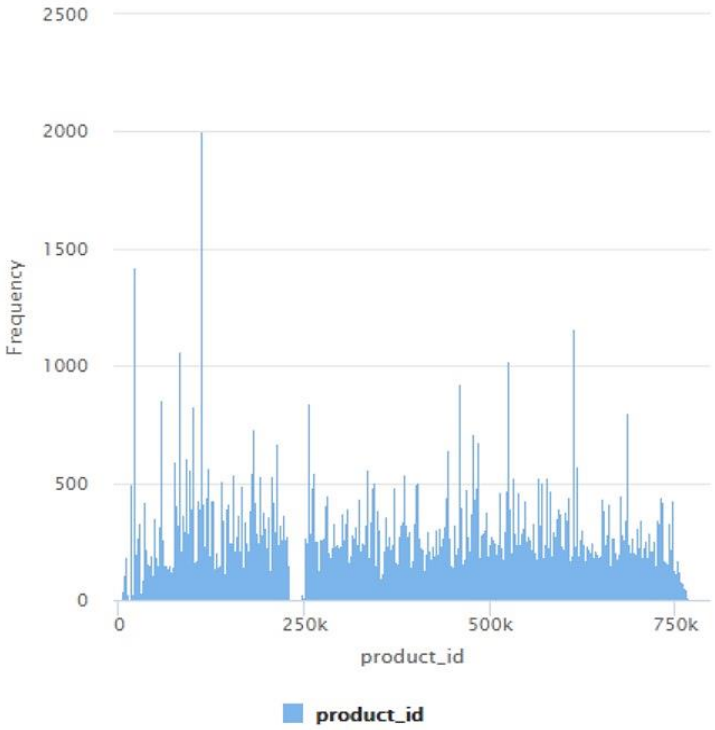

Fig. 7. Frequency of each product sold in data set 3.

3 | Research Methodology

The database of Iranian online shopping companies could be analyzed by the possible methods as follows:

I. Dataset 1: product history anomaly detection, price forecasting, statistical analysis of price stability between categories, and using machine learning to identify price errors by sellers.

II. Dataset 2: user comments natural language processing, classification based on comment quality, spam detection, and psychological analysis.

III. Dataset 3: customer purchase history trend analysis across cities, forecasting customer purchases and orders, and customer categorization.

IV. Dataset 4: quality product reviews anomaly detection, future price forecasting, price statistical analysis and stability across categories, and using machine learning to identify incorrect prices by vendors.

V. Dataset 5: product list classification prediction, anomaly detection, categorization error detection, duplicate detection, and dynamic categorization using data attributes.

3.1 | Preproccessing of the Data

Data preprocessing is a method in data mining that converts raw data into a more practical and effective format. In this work, data preprocessing involves making changes to or removing data before using it to improve performance and is a crucial part of the data mining process. Data collection techniques are frequently not closely monitored, leading to discrepancies such as negative income, unrealistic data pairings (e.g., Male gender, pregnancy status: yes), and missing data. Assessing unvetted data may lead to inaccurate findings. Therefore, the most critical aspect before conducting the analysis is data representation and quality; the most crucial stage of a machine learning project is data preprocessing. Discovering knowledge during the training phase becomes more challenging when there is a large amount of irrelevant or redundant information and noisy and unreliable data. Also, preparing and filtering data can require a significant amount of time for processing.

In this paper, data preprocessing comprises cleaning, normalization, feature extraction, and selection. We have implemented the suitable ones for our dataset. We have cleaned our dataset from missing data and duplicates. We have also considered the following dirty data: misleading data, duplicate data, incorrect data, inaccurate data, non-integrated data, data that violates business rules, data without generalized formatting, and incorrectly punctuated or spelled data.

3.2 | Data Classification Methods for Dataset 3 (Customer Purchase History)

This section has implemented six classification methods, including Random Forest, gradient-boosted trees, K-Nearest Neighbor, Naïve Bayes, Kernel Naïve Bayes, and Neural Network [22], on the third dataset the online shopping company presented. The classification tools are selected because they help segment customers into predefined categories, such as loyal, new, or at-risk, based on purchase frequency, total spending, or product preferences. This enables targeted marketing, improved customer retention, and customized offers, enhancing customer satisfaction and boosting sales. Also, how to code these methods on Rapid Miner Studio, their parameter tuning, and performance metrics are provided. In all classification methods, the label attribute is 'price,' and the other attributes are 'Order id,' 'Product id,' 'State id,' and 'quantity' with about 200000 instances (rows). All classification models are implemented in Rapid Miner Studio software version 9.9 using an operating machine with Processor Intel(R) Core(TM) 1.60GHz properties with 16 GB RAM. Notably, each method's feature selection and hyperparameter tuning were performed on the Rapi Miner software.

3.2.1 | Random Forest

First, the Random Forest classification method has been implemented in Rapid Miner Studio. After reading the data by Rapid Miner, missing values, errors, and untidy data are removed from the dataset. Then, the relevant attributes are selected by eliminating the redundant and highly correlated attributes. Next, the price attribute is discretized since it is the label attribute and must be the nominal type to be acceptable by Random Forest. Also, the parameter tuning for Random Forest is shown in *Fig. 8*. These parameters are obtained according to the least error in the test set. The confusion matrix, accuracy, and other classification metrics for Random Forest are given in *Fig. 9*.

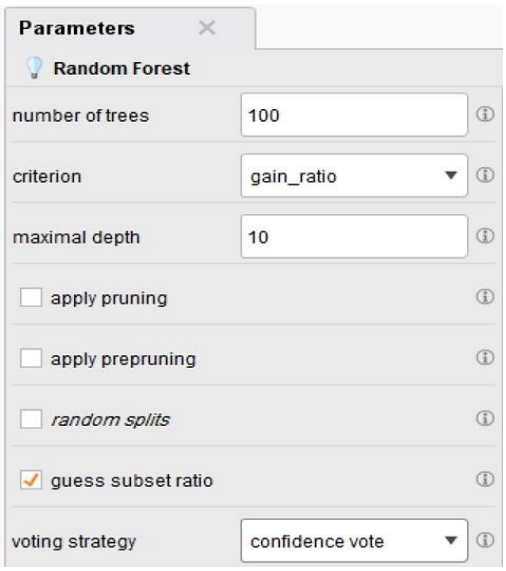

Fig. 8. Parameter tuning for Random Forest in Rapid Miner.

3.2.2| Gradient-boosted trees

The gradient-boosted trees algorithm for the dataset is implemented in Rapid Miner Studio. After reading the data by software, the attributes of city id, state _id, product _id, and order id are transformed from numerical type to nominal. Next, missing values, errors, and untidy data are removed from the dataset. Then, the relevant attributes are selected by eliminating the redundant and highly correlated attributes. Next, the price attribute is discretized since it is the label attribute. Also, the parameter tuning for gradient-boosted trees is shown in *Fig. 10*. The classification metrics for gradient-boosted trees are given in *Fig. 11*. As observed, this dataset's accuracy metric is improper since the class label distribution is imbalanced and must be balanced to verify its results.

```
PerformanceVector:
accuracy: 99.83%
ConfusionMatrix:
True:   range1 [-∞ - 20]        range2 [20 - 40]        range3 [40 - 60]        range4 [60 - 80]        range5 [80 - ∞]
range1 [-∞ - 20]:       39881   56      8       2       1
range2 [20 - 40]:       0       0       0       0       0
range3 [40 - 60]:       0       0       0       0       0
range4 [60 - 80]:       0       0       0       0       0
range5 [80 - ∞]:        0       0       0       0       0
classification_error: 0.17%
ConfusionMatrix:
True:   range1 [-∞ - 20]        range2 [20 - 40]        range3 [40 - 60]        range4 [60 - 80]        range5 [80 - ∞]
range1 [-∞ - 20]:       39881   56      8       2       1
range2 [20 - 40]:       0       0       0       0       0
range3 [40 - 60]:       0       0       0       0       0
range4 [60 - 80]:       0       0       0       0       0
range5 [80 - ∞]:        0       0       0       0       0
absolute_error: 0.003 +/- 0.041
relative_error: 0.34% +/- 4.08%
squared_error: 0.002 +/- 0.041
```

Fig. 9. Different performance metrics for Random Forest.

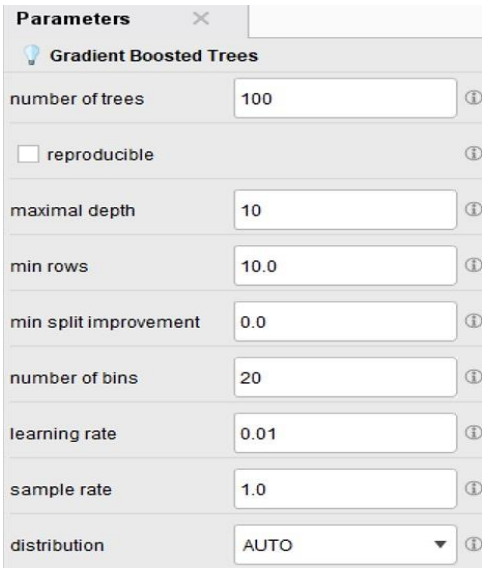

Fig. 10. Parameter tuning for gradient-boosted trees in Rapid Miner.

```
PerformanceVector:
accuracy: 99.83%
ConfusionMatrix:
True:   range1 [-∞ - 20]        range2 [20 - 40]        range3 [40 - 60]        range4 [60 - 80]        range5 [80 - ∞]
range1 [-∞ - 20]:       39881   56      8       2       1
range2 [20 - 40]:       0       0       0       0       0
range3 [40 - 60]:       0       0       0       0       0
range4 [60 - 80]:       0       0       0       0       0
range5 [80 - ∞]:        0       0       0       0       0
classification_error: 0.17%
ConfusionMatrix:
True:   range1 [-∞ - 20]        range2 [20 - 40]        range3 [40 - 60]        range4 [60 - 80]        range5 [80 - ∞]
range1 [-∞ - 20]:       39881   56      8       2       1
range2 [20 - 40]:       0       0       0       0       0
range3 [40 - 60]:       0       0       0       0       0
range4 [60 - 80]:       0       0       0       0       0
range5 [80 - ∞]:        0       0       0       0       0
absolute_error: 0.225 +/- 0.030
relative_error: 22.46% +/- 2.96%
squared_error: 0.051 +/- 0.034
```

Fig. 11. Different performance metrics for Gradient-boosted trees.

3.2.3 | K-Nearest Neigbors (KNN) algorithm

This section implements the KNN classification method in Rapid Miner Studio. First, the database is read by software, and then missing values, errors, and untidy data are removed from the dataset. Then, the relevant attributes are selected by eliminating the redundant and highly correlated attributes. Next, the price attribute is discretized. Finally, the database enters the KNN module. The parameter tuning for KNN is shown in *Fig. 12*, where its essential parameter is K, and the class label is determined with the weighted vote. Also, classification metrics for KNN with K=2, K=5, and K=10 are given in *Figs. 13-15*, respectively.

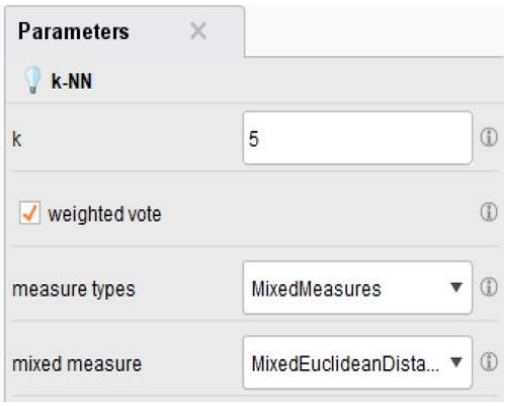

Fig. 12. Parameter tuning for KNN in Rapid Miner.

```
PerformanceVector:
accuracy: 99.81%
ConfusionMatrix:
True:   range1 [-∞ - 20]        range2 [20 - 40]        range3 [40 - 60]        range4 [60 - 80]        range5 [80 - ∞]
range1 [-∞ - 20]:       39871   55      7       2       0
range2 [20 - 40]:       9       1       0       0       1
range3 [40 - 60]:       1       0       1       0       0
range4 [60 - 80]:       0       0       0       0       0
range5 [80 - ∞]:        0       0       0       0       0
classification_error: 0.19%
ConfusionMatrix:
True:   range1 [-∞ - 20]        range2 [20 - 40]        range3 [40 - 60]        range4 [60 - 80]        range5 [80 - ∞]
range1 [-∞ - 20]:       39871   55      7       2       0
range2 [20 - 40]:       9       1       0       0       1
range3 [40 - 60]:       1       0       1       0       0
range4 [60 - 80]:       0       0       0       0       0
range5 [80 - ∞]:        0       0       0       0       0
absolute_error: 0.002 +/- 0.046
relative_error: 0.23% +/- 4.56%
squared_error: 0.002 +/- 0.044
```

Fig. 13. Different performance metrics for KNN with K=2.

```
PerformanceVector:
accuracy: 99.83%
ConfusionMatrix:
True:   range1 [-∞ - 20]        range2 [20 - 40]        range3 [40 - 60]        range4 [60 - 80]        range5 [80 - ∞]
range1 [-∞ - 20]:       39881   56      8       2       1
range2 [20 - 40]:       0       0       0       0       0
range3 [40 - 60]:       0       0       0       0       0
range4 [60 - 80]:       0       0       0       0       0
range5 [80 - ∞]:        0       0       0       0       0
classification_error: 0.17%
ConfusionMatrix:
True:   range1 [-∞ - 20]        range2 [20 - 40]        range3 [40 - 60]        range4 [60 - 80]        range5 [80 - ∞]
range1 [-∞ - 20]:       39881   56      8       2       1
range2 [20 - 40]:       0       0       0       0       0
range3 [40 - 60]:       0       0       0       0       0
range4 [60 - 80]:       0       0       0       0       0
range5 [80 - ∞]:        0       0       0       0       0
absolute_error: 0.003 +/- 0.043
relative_error: 0.26% +/- 4.28%
squared_error: 0.002 +/- 0.041
```

Fig. 14. Different performance metrics for KNN with K=5.

```
PerformanceVector:
accuracy: 99.83%
ConfusionMatrix:
True:   rangel [-∞ - 20]        range2 [20 - 40]        range3 [40 - 60]        range4 [60 - 80]        range5 [80 - ∞]
rangel [-∞ - 20]:       39881   56      8       2       1
range2 [20 - 40]:       0       0       0       0       0
range3 [40 - 60]:       0       0       0       0       0
range4 [60 - 80]:       0       0       0       0       0
range5 [80 - ∞]:        0       0       0       0       0
classification_error: 0.17%
ConfusionMatrix:
True:   rangel [-∞ - 20]        range2 [20 - 40]        range3 [40 - 60]        range4 [60 - 80]        range5 [80 - ∞]
rangel [-∞ - 20]:       39881   56      8       2       1
range2 [20 - 40]:       0       0       0       0       0
range3 [40 - 60]:       0       0       0       0       0
range4 [60 - 80]:       0       0       0       0       0
range5 [80 - ∞]:        0       0       0       0       0
absolute_error: 0.002 +/- 0.041
relative_error: 0.22% +/- 4.14%
squared_error: 0.002 +/- 0.041
```

Fig. 15. Different performance metrics for KNN with K=10.

3.2.4 | Naïve bayes

In this section, the Naïve Bayes classification method is coded in Rapid Miner Studio. The software first reads the database, removing missing values, errors, and untidy data. Then, the relevant attributes are chosen by eliminating redundant and highly correlated attributes. Next, the price attribute is discretized. Finally, the database enters the Naïve Bayes module. The confusion matrix and classification metrics for Naïve Bayes are given in *Fig. 16*.

3.2.5 | Naïve Bayes with Kernal Density Estimation

First, to employ the Naïve Bayes with Kernel Density Estimation (NB-KDE), the database is given to software, and then missing values, errors, and untidy data are removed from the dataset. Then, the relevant attributes are chosen by eliminating the redundant and highly correlated attributes. Next, the price attribute is discretized since it is the label attribute. The parameters of the NB-KDE classification method are tuned, as shown in *Fig. 17*. Finally, the database enters the NB-KDE module. Also, the confusion matrix and classification metrics for NB-KDE with different values of Kernel=2, 5, and 10 are given in *Figs. 18-20*, respectively.

```
PerformanceVector:
accuracy: 85.17%
ConfusionMatrix:
True:   rangel [-∞ - 20]        range2 [20 - 40]        range3 [40 - 60]        range4 [60 - 80]        range5 [80 - ∞]
rangel [-∞ - 20]:       34015   19      1       2       0
range2 [20 - 40]:       845     5       1       0       0
range3 [40 - 60]:       4097    29      4       0       0
range4 [60 - 80]:       889     3       2       0       1
range5 [80 - ∞]:        35      0       0       0       0
classification_error: 14.83%
ConfusionMatrix:
True:   rangel [-∞ - 20]        range2 [20 - 40]        range3 [40 - 60]        range4 [60 - 80]        range5 [80 - ∞]
rangel [-∞ - 20]:       34015   19      1       2       0
range2 [20 - 40]:       845     5       1       0       0
range3 [40 - 60]:       4097    29      4       0       0
range4 [60 - 80]:       889     3       2       0       1
range5 [80 - ∞]:        35      0       0       0       0
absolute_error: 0.155 +/- 0.345
relative_error: 15.51% +/- 34.53%
squared_error: 0.143 +/- 0.343
```

Fig. 16. Different performance metrics for Naïve Bayes.

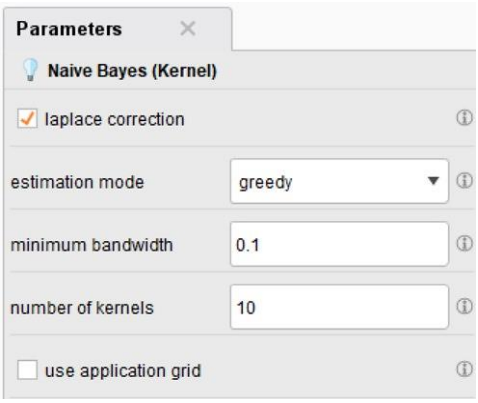

Fig. 17. Parameters of NB-KDE.

accuracy: 85.02%

	true range1 [-∞ - 20]	true range2 [20 - 40]	true range3 [40 - 60]	true range4 [60 - 80]	true range5 [80 - ∞]	class precision
pred. range1 [-∞ - 20]	33954	19	1	2	0	99.94%
pred. range2 [20 - 4...	906	5	2	0	0	0.55%
pred. range3 [40 - 6...	4097	29	3	0	0	0.07%
pred. range4 [60 - 8...	889	3	2	0	1	0.00%
pred. range5 [80 - ∞]	35	0	0	0	0	0.00%
class recall	85.14%	8.93%	37.50%	0.00%	0.00%	

Fig. 18. Performance of Naïve Bayes with Kernel=2 (confusion matrix and accuracy).

accuracy: 85.04%

	true range1 [-∞ - 20]	true range2 [20 - 40]	true range3 [40 - 60]	true range4 [60 - 80]	true range5 [80 - ∞]	class precision
pred. range1 [-∞ - 20]	33942	19	1	2	0	99.94%
pred. range2 [20 - 4...	4433	30	5	0	1	0.67%
pred. range3 [40 - 6...	585	4	0	0	0	0.00%
pred. range4 [60 - 8...	886	3	2	0	0	0.00%
pred. range5 [80 - ∞]	35	0	0	0	0	0.00%
class recall	85.11%	53.57%	0.00%	0.00%	0.00%	

Fig. 19. Performance of Naïve Bayes with Kernel=5 (confusion matrix and accuracy).

accuracy: 84.99%

	true range1 [-∞ - 20]	true range2 [20 - 40]	true range3 [40 - 60]	true range4 [60 - 80]	true range5 [80 - ∞]	class precision
pred. range1 [-∞ - 20]	33922	19	1	2	1	99.93%
pred. range2 [20 - 4...	4452	30	5	0	0	0.67%
pred. range3 [40 - 6...	585	4	0	0	0	0.00%
pred. range4 [60 - 8...	886	3	2	0	0	0.00%
pred. range5 [80 - ∞]	35	0	0	0	0	0.00%
class recall	85.06%	53.57%	0.00%	0.00%	0.00%	

Fig. 20. Performance of Naïve Bayes with Kernel=10 (confusion matrix and accuracy).

3.2.6 | Nueral Network

To employ the Neural Network (NN), it is implemented in Rapid Miner Studio. First, the database is read by software, and then missing values, errors, and untidy data are removed from the dataset. Then, the relevant attributes are selected by eliminating the redundant and highly correlated attributes. Next, the price attribute is discretized. Finally, the database enters the NN module. The parameters of the NN classification method are tuned, as shown in *Fig. 21*. Also, the classification metrics for NN are given in *Fig. 22*.

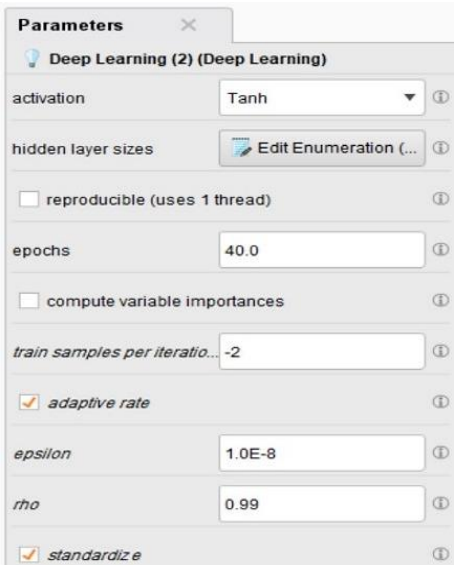

Fig. 21. Parameters of NN in Rapid Miner

```
PerformanceVector:
accuracy: 67.12%
ConfusionMatrix:
True:   rangel [-∞ - 0.000]     range2 [0.000 - 0.000]  range3 [0.000 - 0.000]  range4 [0.000 - 0.000]  range5 [0.000 - ∞]
rangel [-∞ - 0.000]:    0       0       0       0       0
range2 [0.000 - 0.000]: 0       0       0       0       0
range3 [0.000 - 0.000]: 1       0       9       2       7
range4 [0.000 - 0.000]: 0       0       0       1       1
range5 [0.000 - ∞]:     24      0       51      10      186
classification_error: 32.88%
ConfusionMatrix:
True:   rangel [-∞ - 0.000]     range2 [0.000 - 0.000]  range3 [0.000 - 0.000]  range4 [0.000 - 0.000]  range5 [0.000 - ∞]
rangel [-∞ - 0.000]:    0       0       0       0       0
range2 [0.000 - 0.000]: 0       0       0       0       0
range3 [0.000 - 0.000]: 1       0       9       2       7
range4 [0.000 - 0.000]: 0       0       0       1       1
range5 [0.000 - ∞]:     24      0       51      10      186
absolute_error: 0.404 +/- 0.335
relative_error: 40.44% +/- 33.48%
squared_error: 0.276 +/- 0.337
correlation: 0.093
```

Fig. 22. Performance metrics of NN.

3.2.7 | Comparison of Random Forest, gradient-boosted trees, KNN with K=5, Naïve Bayes, and NN

This section evaluates the performance of the different classification methods implemented in the dataset 3 using various criteria. In the previous sections, the accuracy and other performance metrics indicate the superiority of each method against the other one. However, in the earlier sections, as mentioned, the training and test sets are imbalanced, which impacts the preciseness of the models. We use the "Under sampling" balancing method to repair the imbalanced dataset and overcome this issue. So, 75% of the instances related to the majority label of the price attribute are removed, and then the remaining 25% are normalized and

discretized to go to classification mining. Also, after "Under sampling," the outliers are detected and removed to obtain a better model. The comparison of the performance of Random Forest (with 100 trees), Gradient-boosted trees (with 100 trees), KNN (with K=5), Naïve Bayes, and NN (Deep Learning) is given as Receiver Operating Characteristic (ROC) curve (with ten folds and 0.7 as the split ratio) in *Fig. 23* (before balancing) and *Fig. 24* (after balancing). ROC curve plots the "true positive rate" (sensitivity) against the "false positive rate" (1-specificity). A good classifier produces a curve that rises quickly toward the top left corner, indicating high true positives with low false positives. The Area Under the Curve (AUC) summarizes the overall performance, with 1.0 being a perfect model and 0.5 representing random guessing.

These results show that before balancing the dataset, the KNN with K=5 is the best classification method among the existing techniques for dataset 3 since its area below the ROC curve is higher than the others. However, after balancing, gradient-boosted trees outperform the remaining classification methods regarding the area below the ROC curve.

3.3|Clustering Methods for Dataset 3 (Customer Purchase History)

In this section, two clustering methods, K-Means, and X-Means, have been implemented on dataset 3. Clustering methods group customers with similar purchasing behaviors without requiring predefined categories. This allows businesses to discover hidden patterns, such as emerging trends in customer preferences, niche markets, or untapped potential. By understanding these groups, companies can tailor recommendations, optimize product assortments, and personalize marketing strategies, ultimately increasing customer engagement and profitability. Also, how to code these methods on Rapid Miner Studio, their parameter tuning, and performance metrics are provided. In all clustering methods, there is no label attribute, and the main attributes are "price," "Order id," "Product _id," "State _id," and "quantity" with about 200000 instances (rows). All clustering models are implemented in Rapid Miner Studio software version 9.9 using an operating machine with Processor Intel(R) Core(TM) 1.60GHz properties with 16 GB RAM.

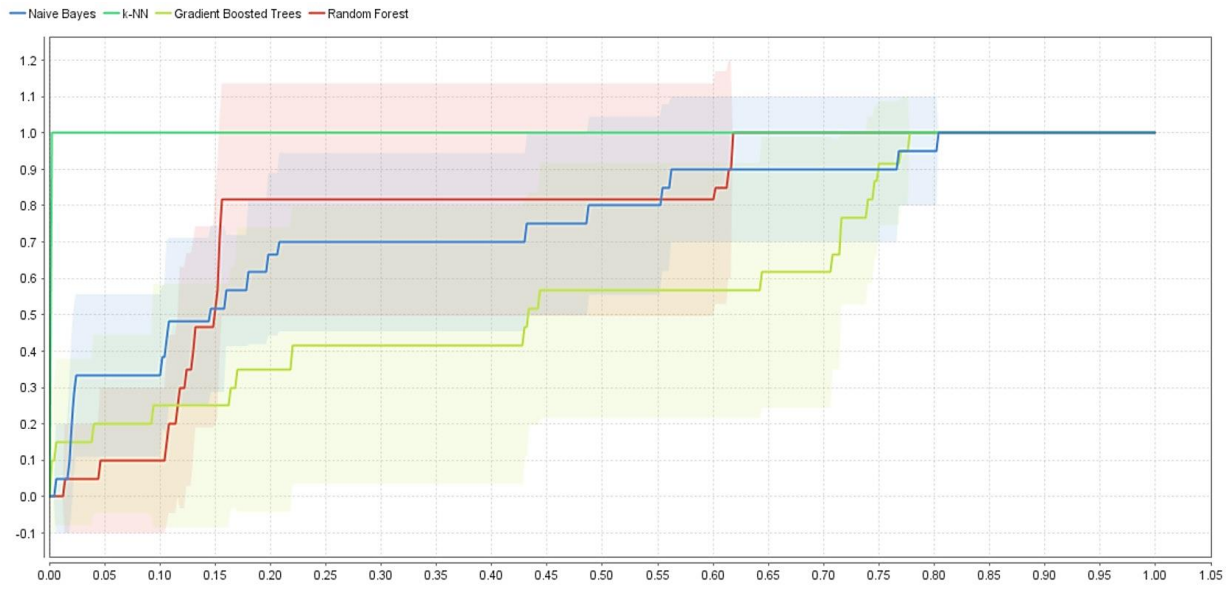

Fig. 23. Comparison of Random Forest, Gradient boosted trees, KNN (K=5), and Naïve Bayes by ROC curve before balancing.

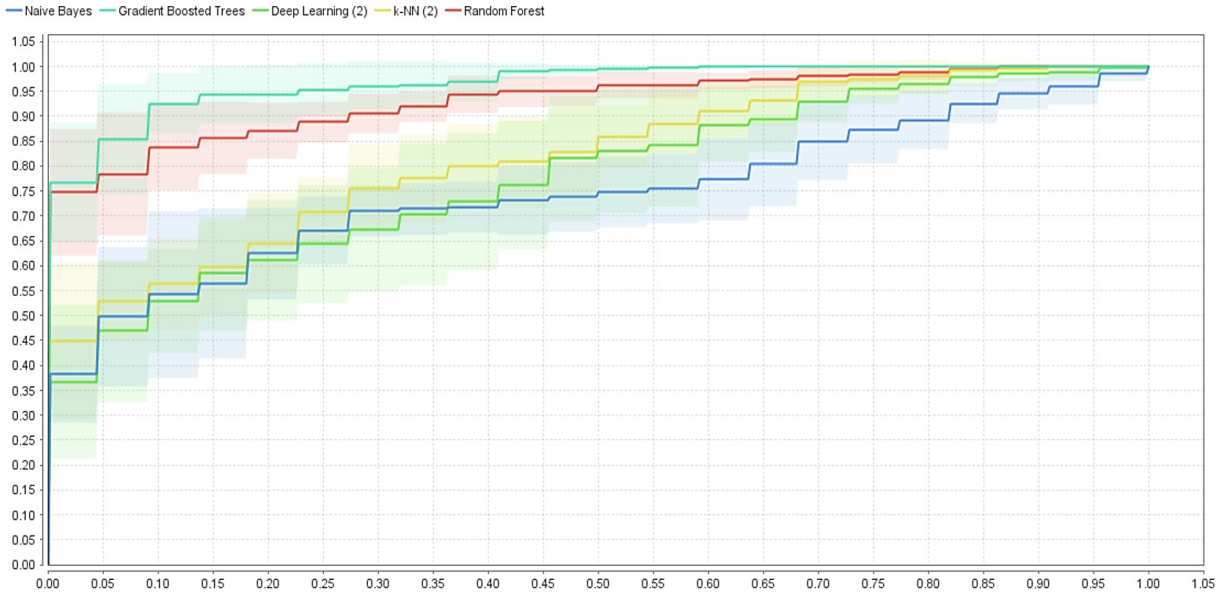

Fig. 24. Comparison of Random Forest, Gradient boosted trees, KNN (K=5), Naïve Bayes, NN (Deep Learning) by ROC curve after balancing and deleting outliers.

3.3.1 | K-Means

First, its modules are coded in Rapid Miner Studio to employ the K-Means method for our dataset. Then, missing values, errors, and untidy data are removed from the dataset. Next, the relevant attributes are selected by eliminating the redundant and highly correlated attributes. Here, unlike the classification methods, there is no discretization. Finally, the database enters the K-Means module. The parameters of the K-Means method are tuned, as shown in *Fig. 25*.

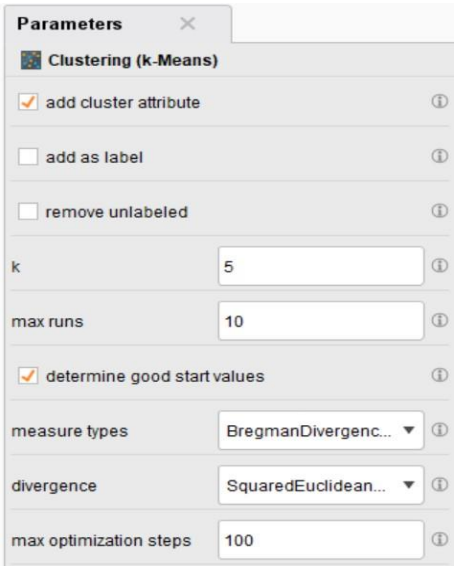

Fig. 25. Parameter tuning for K-Means in Rapid Miner.

3.3.2 | Performance of K-Means with different K's

In this section, the "distance metric" is considered the clustering metric to compare the K-Means with different K's (for K=2, 3, 5, 7, 10, 12, 15, and 20) as given in *Figs. 26-33*. Lower average values within centroid distance for each cluster and" Davis Bouldin" (usually lower) are performance measures for finding the proper value of K in the K-Means clustering method for dataset 3. These results show that when K is three, it is more efficient regarding the average within centroid distance for each cluster. Also, setting K higher than ten does not help improve the K-Means' performance.

```
PerformanceVector:
Avg. within centroid distance: -0.010
Avg. within centroid distance_cluster_0: -0.024
Avg. within centroid distance_cluster_1: -0.007
Davies Bouldin: -0.349
```

Fig. 26. Performance of clustering with K-Means (K=2).

```
PerformanceVector:
Avg. within centroid distance: -0.006
Avg. within centroid distance_cluster_0: -0.008
Avg. within centroid distance_cluster_1: -0.005
Avg. within centroid distance_cluster_2: -0.009
Davies Bouldin: -0.454
```

Fig. 27. Performance of clustering with K-Means (K=3).

```
PerformanceVector:
Avg. within centroid distance: -0.001
Avg. within centroid distance_cluster_0: -0.000
Avg. within centroid distance_cluster_1: -0.002
Avg. within centroid distance_cluster_2: -0.005
Avg. within centroid distance_cluster_3: -0.002
Avg. within centroid distance_cluster_4: -0.011
Avg. within centroid distance_cluster_5: -0.001
Avg. within centroid distance_cluster_6: -0.023
Avg. within centroid distance_cluster_7: -0.002
Avg. within centroid distance_cluster_8: -0.023
Avg. within centroid distance_cluster_9: -0.002
Davies Bouldin: -0.616
```

Fig. 28. Performance of clustering with K-Means (K=5).

```
PerformanceVector:
Avg. within centroid distance: -0.003
Avg. within centroid distance_cluster_0: -0.002
Avg. within centroid distance_cluster_1: -0.006
Avg. within centroid distance_cluster_2: -0.016
Avg. within centroid distance_cluster_3: -0.007
Avg. within centroid distance_cluster_4: -0.003
Davies Bouldin: -0.553
```

Fig. 29. Performance of clustering with K-Means (K=7).

```
PerformanceVector:
Avg. within centroid distance: -0.002
Avg. within centroid distance_cluster_0: -0.001
Avg. within centroid distance_cluster_1: -0.004
Avg. within centroid distance_cluster_2: -0.006
Avg. within centroid distance_cluster_3: -0.002
Avg. within centroid distance_cluster_4: -0.011
Avg. within centroid distance_cluster_5: -0.004
Avg. within centroid distance_cluster_6: -0.016
Davies Bouldin: -0.590
```

Fig. 30. Performance of clustering with K-Means (K=10).

```
PerformanceVector:
Avg. within centroid distance: -0.001
Avg. within centroid distance_cluster_0: -0.001
Avg. within centroid distance_cluster_1: -0.000
Avg. within centroid distance_cluster_2: -0.010
Avg. within centroid distance_cluster_3: -0.016
Avg. within centroid distance_cluster_4: -0.002
Avg. within centroid distance_cluster_5: -0.001
Avg. within centroid distance_cluster_6: -0.002
Avg. within centroid distance_cluster_7: -0.097
Avg. within centroid distance_cluster_8: -0.002
Avg. within centroid distance_cluster_9: -0.008
Avg. within centroid distance_cluster_10: -0.001
Avg. within centroid distance_cluster_11: -0.002
Avg. within centroid distance_cluster_12: -0.012
Avg. within centroid distance_cluster_13: -0.010
Avg. within centroid distance_cluster_14: -0.001
Davies Bouldin: -0.615
```

Fig. 31. Performance of clustering with K-Means (K=12).

```
PerformanceVector:
Avg. within centroid distance: -0.001
Avg. within centroid distance_cluster_0: -0.001
Avg. within centroid distance_cluster_1: -0.002
Avg. within centroid distance_cluster_2: -0.002
Avg. within centroid distance_cluster_3: -0.000
Avg. within centroid distance_cluster_4: -0.003
Avg. within centroid distance_cluster_5: -0.002
Avg. within centroid distance_cluster_6: -0.009
Avg. within centroid distance_cluster_7: -0.001
Avg. within centroid distance_cluster_8: -0.024
Avg. within centroid distance_cluster_9: -0.002
Avg. within centroid distance_cluster_10: -0.010
Avg. within centroid distance_cluster_11: -0.022
Davies Bouldin: -0.561
```

Fig. 32. Performance of clustering with K-Means (K=15).

```
PerformanceVector:
Avg. within centroid distance: -0.000
Avg. within centroid distance_cluster_0: -0.001
Avg. within centroid distance_cluster_1: -0.000
Avg. within centroid distance_cluster_2: -0.001
Avg. within centroid distance_cluster_3: -0.000
Avg. within centroid distance_cluster_4: -0.001
Avg. within centroid distance_cluster_5: -0.019
Avg. within centroid distance_cluster_6: -0.001
Avg. within centroid distance_cluster_7: -0.000
Avg. within centroid distance_cluster_8: -0.010
Avg. within centroid distance_cluster_9: -0.009
Avg. within centroid distance_cluster_10: -0.001
Avg. within centroid distance_cluster_11: -0.004
Avg. within centroid distance_cluster_12: -0.012
Avg. within centroid distance_cluster_13: -0.001
Avg. within centroid distance_cluster_14: -0.009
Avg. within centroid distance_cluster_15: -0.008
Avg. within centroid distance_cluster_16: -0.007
Avg. within centroid distance_cluster_17: -0.001
Avg. within centroid distance_cluster_18: -0.061
Avg. within centroid distance_cluster_19: -0.000
Davies Bouldin: -0.586
```

Fig. 33. Performance of clustering with K-Means (K=20).

3.3.3 | X-means

To implement the X-means in Rapid Miner, the database is first read by software, and then missing values, errors, and untidy data are removed from the dataset. Then, the relevant attributes are chosen by eliminating

the redundant and highly correlated attributes. Here, unlike the classification methods, there is no need for discretization. Finally, the database enters the X-Means module. The parameters of the X-Means method are tuned, as shown in *Fig. 34*. Finally, the performance metric for X-Means is given in *Fig. 35*, which shows that the optimal number of clusters is 3 in this dataset.

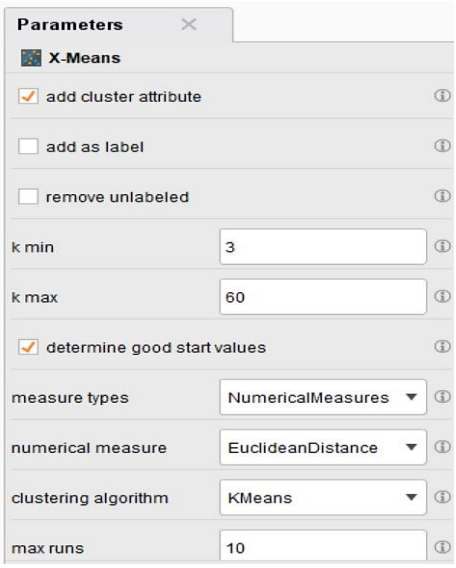

Fig. 34. Parameters of clustering with X-Means.

```
PerformanceVector:
Avg. within centroid distance: -0.006
Avg. within centroid distance_cluster_0: -0.010
Avg. within centroid distance_cluster_1: -0.010
Avg. within centroid distance_cluster_2: -0.004
Davies Bouldin: -0.529
```

Fig. 35. Performance of clustering with X-Means.

4 | Discovered Patterns and Obtained Knowledge

This section provides the obtained knowledge and discovered patterns using the classification and clustering methods. Most results and obtained models cannot be presented due to their largeness and high complexity (especially decision trees and clusters). Still, this work presents some results to show that the models are successfully obtained and can be used as a decision-making tool for any decisionmaker (full results can be shared as the Rapid Miner outputs if requested). *Figs. 36-37* shows a part of the obtained decision trees by the gradient-boosted tree. For example, in *Fig. 37*, this decision tree tells that if the quantity attribute is less than 1.5, the data attribute is among a specific pool. The data attribute is among the other specific pools; then, it is concluded that the price label will be 0.040 (note that the price is normalized). Also, *Fig. 38* shows a part of the obtained results by NN, which shows that, for example, NN predicts the class label 5 for row 21 with a confidence of 0.894.

Moreover *Figs. 39-40* show the visualization of the model obtained by K-Means with K=3 and K=5 when data points are projected to price-state _id axes. Therefore, all discovered patterns can be used as a prediction tool for the same kind of data.

According to the results obtained from the models, utilizing KNN with a K value of 5 before balancing the dataset can assist businesses in acquiring a detailed insight into customer preferences and behavior trends. This method is advantageous for capturing differences in local data and recognizing key factors that impact purchase choices. For organizations with unbalanced datasets, this approach offers a more precise

representation of the core customer segments and assists in customizing marketing strategies. Additionally, the outstanding results observed with gradient-boosted trees after dataset balancing indicate that these techniques successfully forecast customer behavior and purchasing trends. E-commerce companies should consider incorporating gradient-boosted trees into their analysis systems to improve predictive accuracy, resulting in better targeting and increased customer satisfaction. Furthermore, the K-Means clustering results show that customer segmentation based on their purchase history is most meaningful when using a moderate number of clusters (K=3). This understanding can assist e-commerce businesses in creating focused tactics for varying customer groups, resulting in better resource management and personalized marketing initiatives.

```
quantity >= 1.500
|   state_id in {10,11,12,13,14,15,16,... (22 more)}: 0.040 {}
|   state_id not in {10,11,12,13,14,15,16,... (22 more)}
|   |   state_id in {9}
|   |   |   date in {2018-08-12 06:19:19,... (11660 more)}
|   |   |   |   date in {2018-08-12 06:19:19,... (11592 more)}
|   |   |   |   |   date in {2018-08-12 06:19:19,... (8617 more)}: 0.040 {}
|   |   |   |   |   date not in {2018-08-12 06:19:19,... (8617 more)}
|   |   |   |   |   |   date in {2018-08-12 06:19:19,... (8661 more)}: 0.035 {}
|   |   |   |   |   |   date not in {2018-08-12 06:19:19,... (8661 more)}
|   |   |   |   |   |   |   quantity < 4.500: 0.040 {}
|   |   |   |   |   |   |   quantity >= 4.500
|   |   |   |   |   |   |   |   date in {2018-08-12 06:19:19,... (10640 more)}
|   |   |   |   |   |   |   |   |   date in {2018-08-12 06:19:19,... (10416 more)}: 0.040 {}
|   |   |   |   |   |   |   |   |   date not in {2018-08-12 06:19:19,... (10416 more)}: 0.035 {}
|   |   |   |   |   |   |   |   date not in {2018-08-12 06:19:19,... (10640 more)}: 0.040 {}
|   |   |   |   |   date not in {2018-08-12 06:19:19,... (11592 more)}: 0.037 {}
|   |   |   date not in {2018-08-12 06:19:19,... (11660 more)}
|   |   |   |   date in {2018-08-12 06:19:19,... (66956 more)}
|   |   |   |   |   date in {2018-08-12 06:19:19,... (48221 more)}
|   |   |   |   |   |   date in {2018-08-12 06:19:19,... (48150 more)}
|   |   |   |   |   |   |   date in {2018-08-12 06:19:19,... (29517 more)}
|   |   |   |   |   |   |   |   date in {2018-08-12 06:19:19,... (29465 more)}
|   |   |   |   |   |   |   |   |   date in {2018-08-12 06:19:19,... (29118 more)}: 0.040 {}
|   |   |   |   |   |   |   |   |   date not in {2018-08-12 06:19:19,... (29118 more)}: 0.040 {}
|   |   |   |   |   |   |   |   date not in {2018-08-12 06:19:19,... (29465 more)}: 0.035 {}
|   |   |   |   |   |   |   date not in {2018-08-12 06:19:19,... (29517 more)}: 0.040 {}
```

Fig. 36. Part of one of the decision trees in gradient boosted trees.

Tree

```
quantity < 1.500
|   date in {2018-08-12 06:19:19,... (68 more)}
|   |   date in {2018-08-12 06:19:19,... (63 more)}: 0.040 {}
|   |   date not in {2018-08-12 06:19:19,... (63 more)}: 0.025 {}
|   date not in {2018-08-12 06:19:19,... (68 more)}
|   |   state_id in {10,14,15,18,19,2,20,... (13 more)}
|   |   |   state_id in {10,15,19,2,21,23,3,6,... (1 more)}
|   |   |   |   date in {2018-08-12 06:19:19,... (68918 more)}
|   |   |   |   |   date in {2018-08-12 06:19:19,... (19041 more)}
|   |   |   |   |   |   date in {2018-08-12 06:19:19,... (19004 more)}
|   |   |   |   |   |   |   state_id in {10,15,2,21,23,3}: 0.040 {}
|   |   |   |   |   |   |   state_id not in {10,15,2,21,23,3}
|   |   |   |   |   |   |   |   date in {2018-08-12 06:19:19,... (1636 more)}
|   |   |   |   |   |   |   |   |   date in {2018-08-12 06:19:19,... (1510 more)}: 0.040 {}
|   |   |   |   |   |   |   |   |   date not in {2018-08-12 06:19:19,... (1510 more)}: 0.035 {}
|   |   |   |   |   |   |   |   date not in {2018-08-12 06:19:19,... (1636 more)}
|   |   |   |   |   |   |   |   |   date in {2018-08-12 06:19:19,... (12608 more)}: 0.040 {}
|   |   |   |   |   |   |   |   |   date not in {2018-08-12 06:19:19,... (12608 more)}: 0.040 {}
|   |   |   |   |   |   date not in {2018-08-12 06:19:19,... (19004 more)}: 0.033 {}
|   |   |   |   |   date not in {2018-08-12 06:19:19,... (19041 more)}
|   |   |   |   |   |   state_id in {10,19,2,21,3,8}: 0.040 {}
|   |   |   |   |   |   state_id not in {10,19,2,21,3,8}
|   |   |   |   |   |   |   date in {2018-08-12 06:19:19,... (55953 more)}
|   |   |   |   |   |   |   |   date in {2018-08-12 06:19:19,... (34260 more)}
```

Fig. 37. Part of one of the decision trees in gradient-boosted trees.

Row No.	price	prediction(p...	confidence(...	confidence(...	confidence(...	confidence(...	confidence(...	order_id	product_id	quantity
21	range5 [0.00...	range5 [0.00...	0.001	0.000	0.072	0.033	0.894	0.872	0.399	0.040
22	range5 [0.00...	range5 [0.00...	0.001	0.000	0.072	0.033	0.894	0.872	0.399	0.040
23	range5 [0.00...	range5 [0.00...	0.001	0.000	0.072	0.033	0.894	0.872	0.399	0.040
24	range5 [0.00...	range5 [0.00...	0.001	0.000	0.072	0.033	0.894	0.872	0.399	0.040
25	range5 [0.00...	range3 [0.00...	0.005	0.000	0.722	0.071	0.201	0.861	0.693	0.020
26	range5 [0.00...	range5 [0.00...	0.003	0.000	0.124	0.010	0.862	0.858	0.750	0.010
27	range4 [0.00...	range5 [0.00...	0.005	0.000	0.219	0.026	0.749	0.852	0.693	0.010
28	range5 [0.00...	range3 [0.00...	0.008	0.000	0.668	0.116	0.208	0.849	0.749	0.010
29	range5 [0.00...	range5 [0.00...	0.019	0.000	0.167	0.040	0.774	0.824	0.604	0
30	range3 [0.00...	range3 [0.00...	0.001	0.000	0.765	0.028	0.206	0.817	0.693	0.141
31	range3 [0.00...	range3 [0.00...	0.001	0.000	0.765	0.028	0.206	0.817	0.693	0.141
32	range1 [-∞ - 0...	range5 [0.00...	0.017	0.000	0.221	0.111	0.651	0.799	0.462	0
33	range5 [0.00...	range5 [0.00...	0.012	0.000	0.144	0.075	0.769	0.799	0.399	0

Fig. 38. A part of NN output in Rapid Miner.

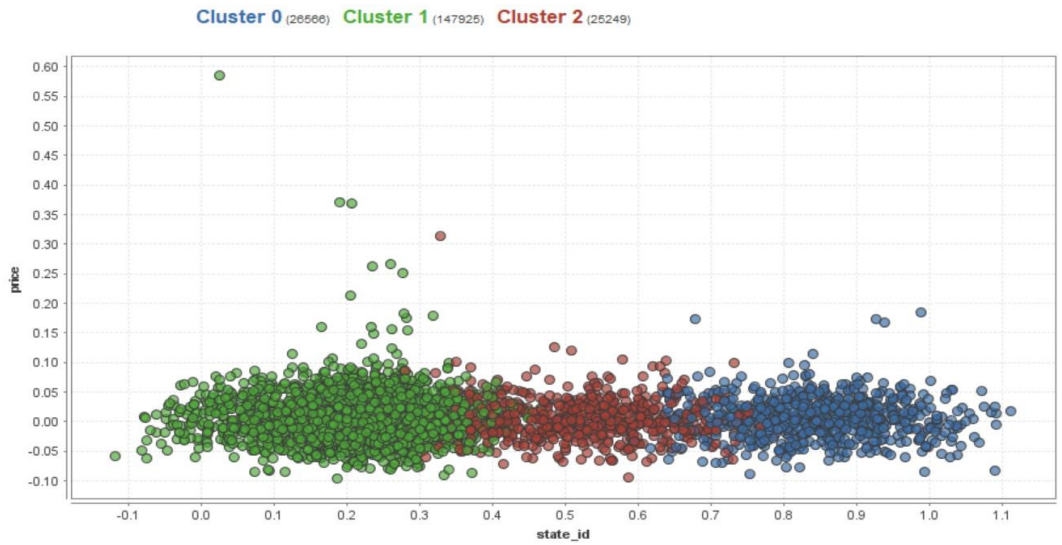

Fig. 39. Visualization of K-Means with K=3 for dataset 3 when data points are projected to price-state id axes.

5 | Conclusion

This paper analyzes the price data of an online shopping company, the first and the most extensive online shop in Iran. This company has more than 17 million daily visits for more than 4 million goods. Such a vast e-commerce company, with this immense amount of data collected each day, needs to utilize proper data mining and machine learning methods to improve its business. The customer purchase history, which includes 200 thousand records of one-week transactions, is selected among the available datasets. Classification and clustering methods have been implemented to discover the knowledge and patterns in the dataset. The results of different classification methods show that before balancing the dataset, the KNN with K=5 is the best classification method among the existing methods. The reason is that KNN is sensitive to the distribution of the data. With K=5, it considers the majority vote among the five closest neighbors, which may have naturally fit the underlying structure of the dataset, especially if the data points are moderately imbalanced. Also, KNN is a local learning algorithm, meaning it makes predictions based on nearby points. With K=5, it balances sensitivity to outliers (which smaller KKK values might suffer from) and smoothing (larger KKK values can overgeneralize). This value could have worked well in distinguishing classes in the unbalanced data. However, after balancing, gradient-boosted trees outperform the remaining classification methods regarding the area

below the ROC curve. For clustering methods, the results show that K Means with K=3 is more efficient regarding the average within centroid distance for each cluster. Also, setting K higher than ten does not help improve the K-Means' performance. All obtained models and discovered patterns can be used as a prediction tool for the same kind of data.

The findings from this study hold significant implications for e-commerce businesses aiming to leverage data-driven strategies to enhance their operations. Implementing KNN with K=5 before dataset balancing can help companies gain a nuanced understanding of customer preferences and behavior patterns. This approach is beneficial for capturing local data variations and identifying key features influencing purchase decisions. For companies with imbalanced datasets, this method provides a more accurate reflection of the underlying customer segments and helps tailor personalized marketing strategies. Also, the superior performance of gradient-boosted trees post-dataset balancing suggests that these methods are highly effective for predicting customer behavior and purchase patterns. E-commerce businesses should consider integrating gradient-boosted trees into their analytics frameworks to enhance predictive accuracy, leading to more effective targeting and improved customer satisfaction. In addition, the K-Means clustering results indicate that a moderate number of clusters (K=3) provides the most meaningful segmentation of customers based on their purchase history. This insight can guide e-commerce companies in developing targeted strategies for different customer segments, leading to more efficient resource allocation and tailored marketing campaigns.

While gradient-boosted trees performed well in this study, exploring other advanced classification algorithms, such as deep learning or ensemble methods, could enhance predictive accuracy. Future research could investigate the applicability of these methods in different e-commerce contexts or with more diverse datasets. Also, in future studies, one can analyze the data using other data mining methods, like Support Vector Machine (SVM) [23], Density-Based Spatial Clustering of Applications with Noise (DBSCAN), or multi-methods [24], and compare their performance with the approaches presented in this project. One can implement data mining tools on the other datasets (quality product reviews or product history) provided by the Iranian company. Also, the datasets related to different industries, like the Game industry [25] and healthcare systems [26], may be analyzed by data mining tools.

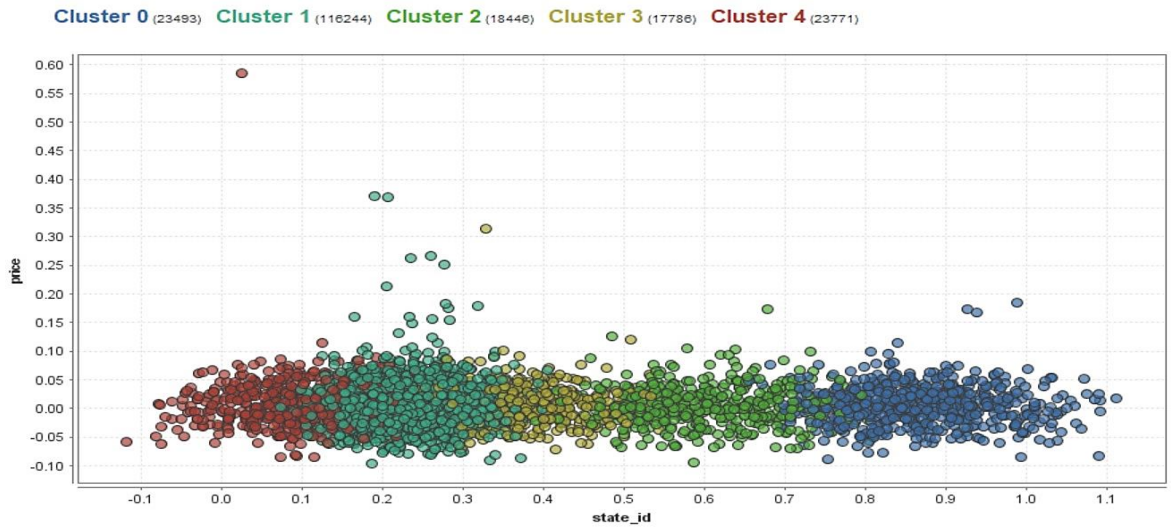

Fig. 40. Visualization of K-Means with K=5 for dataset 3 when data points are projected to price-state_id axes.

Author Contribution

Conceptualization, N.M. and M.J.; Methodology, N.M. and M.J.; Software, N.M. and M.J.; Validation, N.M., M.J.; Formal analysis, N.M. and M.J; investigation, N.M. and M.J.; resources, N.M. and M.J.; data maintenance, M.J.; writing-creating the initial design, N.M. and M.J.; writing-reviewing and editing, N.M.;

visualization, N.M. and M.J.; monitoring, N.M.; project management, N.M. and M.J. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

The authors thank the anonymous reviewers for their valuable comments and suggestions throughout the paper's review process.

Funding

The authors received no financial support for this paper's research, authorship, and publication.

Data Availability

Data, models, and codes are available upon a reasonable request.

Conflicts of Interest

The authors declare no conflict of interest.

Ethical Considerations

This research adhered to high ethical standards by anonymizing customer data to protect privacy and implementing robust security measures. Data was used exclusively for research, with efforts made to identify and mitigate any biases.

References

[1] Siegler, M. A., Jain, U., Raj, B., & Stern, R. M. (1997). *Automatic segmentation, classification and clustering of broadcast news audio* [presentation]. Proceeding darpa speech recognition workshop (Vol. 1997, pp. 97–99). https://www.cs.cmu.edu/~robust/Papers/darpa97_H4-SEG.pdf

[2] Chang, H. J., Hung, L. P., & Ho, C. L. (2007). An anticipation model of potential customers' purchasing behavior based on clustering analysis and association rules analysis. *Expert systems with applications*, *32*(3), 753–764. DOI:10.1016/j.eswa.2006.01.049

[3] Kim, E., & Hong, T. (2010). Segmenting customers in online stores from factors that affect the customer's intention to purchase. *2010 international conference on information society* (pp. 383–388). IEEE. DOI: 10.1109/i-Society16502.2010.6018733

[4] Ponyiam, P., & Arch-int, S. (2018). Customer behavior analysis using data mining techniques. *2018 international seminar on application for technology of information and communication* (pp. 549–554). IEEE. DOI: 10.1109/ISEMANTIC.2018.8549803

[5] Liao, S. H., Chen, Y. J., & Lin, Y. T. (2011). Mining customer knowledge to implement online shopping and home delivery for hypermarkets. *Expert systems with applications*, *38*(4), 3982–3991. DOI:10.1016/j.eswa.2010.09.059

[6] Ait, R., Amine, A., Bouikhalene, B., & Lbibb, R. (2015). Customer segmentation model in e-commerce using clustering techniques and LRFM model : the case of online stores in Morocco. *International journal of computer and information engineering*, *9*(8), 1976–1986.

[7] Ahmeda, R. A. E. D., Shehaba, M. E., Morsya, S., & Mekawiea, N. (2015). Performance study of classification algorithms for consumer online shopping attitudes and behavior using data mining. *2015 fifth international conference on communication systems and network technologies* (pp. 1344–1349). IEEE. DOI: 10.1109/CSNT.2015.50

[8] Gull, M., & Pervaiz, A. (2018). Customer behavior analysis towards online shopping using data mining. *2018 5th international multi-topic ict conference (IMTIC)* (pp. 1–5). IEEE. DOI: 10.1109/IMTIC.2018.8467262

[9] Maheswari, K., & Priya, P. P. A. (2017). Predicting customer behavior in online shopping using svm classifier. *2017 IEEE international conference on intelligent techniques in control, optimization and signal processing (INCOS)* (pp. 1–5). IEEE. DOI: 10.1109/ITCOSP.2017.8303085

[10] Chen, D., Sain, S. L., & Guo, K. (2012). Data mining for the online retail industry: a case study of RFM model-based customer segmentation using data mining. *Journal of database marketing and customer strategy management*, *19*(3), 197–208. DOI:10.1057/dbm.2012.17

[11] Alghanam, O. A., Al-Khatib, S. N., & Hiari, M. O. (2022). Data mining model for predicting customer purchase behavior in e-commerce context. *International journal of advanced computer science and applications*, *13*(2), 421–428. DOI:10.14569/IJACSA.2022.0130249

[12] Moon, N. N., Talha, I. M., & Salehin, I. (2021). An advanced intelligence system in customer online shopping behavior and satisfaction analysis. *Current research in behavioral sciences*, *2*, 100051. DOI:10.1016/j.crbeha.2021.100051

[13] Kazemi, A., Babaei, M. E., & Javad, M. O. M. (2015). A data mining approach for turning potential customers into real ones in basket purchase analysis. *International journal of business information systems*, *19*(2), 139–158. DOI:10.1504/IJBIS.2015.069427

[14] Anbalagan, E., Mohan, E., Puttamadappa, C., Sudhaakar, K., & Saravanan, D. (2009). Building e-shop using incremental association rule. *International journal of computational intelligence research*, *5*(1), 11–23.

[15] Suchacka, G., & Chodak, G. (2017). Using association rules to assess purchase probability in online stores. *Information systems and e-business management*, *15*(3), 751–780. DOI:10.1007/s10257-016-0329-4

[16] Khasanah, A. U., Wibowo, K. S., & Dewantoro, H. F. (2017). The application of data mining techniques to create promotion strategy for mobile phone shop. *IOP conference series: materials science and engineering* (Vol. 277, p. 12013). IOP Publishing. DOI: 10.1088/1757-899X/277/1/012013

[17] Riaz, M., Arooj, A., Hassan, M. T., & Kim, J. B. (2014). Clustering based association rule mining on online stores for optimized cross product recommendation. *The 2014 international conference on control, automation and information sciences (ICCAIS 2014)* (pp. 176–181). IEEE. DOI: 10.1109/ICCAIS.2014.7020553

[18] Hidayat, A. A., Rahman, A., Wangi, R. M., Abidin, R. J., Fuadi, R. S., & Budiawan, W. (2019). Implementation and comparison analysis of apriori and fp-growth algorithm performance to determine market basket analysis in breiliant shop. *Journal of physics: conference series* (Vol. 1402, p. 77031). IOP Publishing. DOI: 10.1088/1742-6596/1402/7/077031

[19] Samboteng, L., Rulinawaty, Kasmad, M. R., Basit, M., & Rahim, R. (2022). Market basket analysis of administrative patterns data of consumer purchases using data mining technology. *Journal of applied engineering science*, *20*(2), 339–345. DOI:10.5937/jaes0-32019

[20] Imani, A., Abbasi, M., Ahang, F., Ghaffari, H., & Mehdi, M. (2022). Customer segmentation to identify key customers based on RFM model by using data mining techniques. *International journal of research in industrial engineering*, *11*(1), 62–76. https://www.riejournal.com/article_138379.html%0Ahttps://www.riejournal.com/article_138379_2db2dd42423125ffd041fef6e313c9a0.pdf

[21] Chiang, W. Y. (2018). Applying data mining for online CRM marketing strategy: an empirical case of coffee shop industry in Taiwan. *British food journal*, *120*(3), 665–675. DOI:10.1108/BFJ-02-2017-0075

[22] Aftabi, N., Moradi, N., & Mahroo, F. (2024). *Feed-forward neural networks as a mixed-integer program*. ArXiv Preprint ArXiv:2402.06697.

[23] Barzegar, M., & Hasani, A. (2024). Analyzing customer churn behavior using datamining approach: hybrid support vector machine and logistic regression in retail chain. *International journal of research in industrial engineering*, *13*(4), 384–398.

[24] Aftabi, N., Moradi, N., Mahroo, F., & Kianfar, F. (2024). *A multi-method framework for information security investment*. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4730222

[25] Moradi, N., Malekmohammad, H., & Jamalzadeh, S. (2018). A model for performance evaluation of digital game industry using integrated AHP and BSC. *Journal of applied research on industrial engineering*, *5*(2), 97–109. https://www.researchgate.net/publication/326904946_A_Model_for_Performance_Evaluation_of_Digital_Game_Industry_using_Integrated_AHP_and_BSC

[26] Mostofi, S., Kordrostami, S., Refahi Shekhani, A., Faridi Masouleh, M., & Shokri, S. (2022). Data mining and diagnosis of heart diseases: a hybrid approach to the b-mine algorithm and association rules. *International journal of research in industrial engineering*, *11*(1), 77–91.